CTJ: INPUT-OUTPUT BASED RELATION COMBINATORIAL TESTING STRATEGY USING JAYA ALGORITHM

NG YEONG KHANG

BACHELOR OF COMPUTER SCIENCE
(SOFTWARE ENGINEERING)

UNIVERSITI MALAYSIA PAHANG

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Bachelor of Computer Science (Software Engineering).

_____

(Supervisor's Signature)

Full Name : Dr. AbdulRahman al-Sewari

Position : Senior Lecturer

Date : 8 January 2019

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

_____

(Student's Signature)

Full Name : Ng Yeong Khang

ID Number : CB15092

Date : 8 January 2019

CTJ: INPUT-OUTPUT BASED RELATION COMBINATORIAL TESTING STRATEGY USING JAYA ALGORITHM

NG YEONG KHANG

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Bachelor of Computer Science (Software Engineering)

Faculty of Computer Systems and Software Engineering

UNIVERSITI MALAYSIA PAHANG

JANUARY 2019

ACKNOWLEDGEMENTS

First, I would express my deepest appreciation to my respected final year project supervisor, Dr. AbdulRahman al-Sewari in guiding me to complete this thesis which has the title of CTJ: Input-Output Based Relation Combinatorial Testing Strategy Using Jaya Algorithm. He always welcomes me to consult him anytime whenever I faced any problem in doing this final year project. His expertise in software testing gave me insight and well understanding on combinatorial testing and this knowledge accelerates my progress in completing this thesis.

Moreover, I would like to thank to my family members especially my dear parents in supporting me throughout the implementation of my final year project. They always give me advice and backing when I was no idea to solve the trouble encountered. The strong mental support given by them is a great motivation for me to complete this thesis.

Furthermore, a special thanks goes to all friends of mine who gave their helping hands when I need any assistance from them. Their willingness in helping me should be acknowledged as their contributions made the progress of my final year project run uneventful.

Finally, a special gratitude I want to give to all lecturers and staff in Faculty of Computer Systems and Software Engineering, Universiti Malaysia Pahang that helped me directly and indirectly in completing this final year project. Their precious helps made me able to work smoothly throughout this final year project.

ABSTRAK

Pengujian perisian adalah salah satu unsur yang penting dalam pembangunan perisian. Kebanyakan masa, sistem yang diuji mempunyai lebih daripada satu input dan pengujian setiap kombinasi input adalah hampir mustahil kerana masa pelaksanaan kes ujian terlalu panjang. Pengujian kombinatorial ialah satu cara untuk menggantikan ujian menyeluruh melalui pengujian setiap nilai input dan setiap kombinasi antara parameter. Pengujian kombinatorial boleh dibahagikan kepada tiga jenis iaitu interaksi kekuatan seragam, interaksi kekuatan berubah-ubah dan hubungan berdasarkan input-output (IOR). Pengujian kombinatorial IOR hanya menguji kombinasi penting yang dipilih oleh penguji. Kebanyakan penyelidikan dalam pengujian kombinatorial menggunakan interaksi kekuatan seragam dan berubah-ubah tetapi terdapat hanya beberapa kajian yang menangani IOR. Oleh hal sedemikian, pengujian kombinatorial IOR dipilih untuk dikaji dalam kajian ini. Untuk mengatasi masalah pengoptimalan gabungan, algoritma Jaya dicadangkan untuk digunakan dalam projek ini disebabkan algoritma metaheuristik pantas dalam pengoptimuman dan strategi ini dinamakan sebagai CTJ. Hasil penerapan algoritma Jaya dalam pegujian kombinatorial input-output dapat diterima kerana menghasilkan jumlah kes ujian yang hampir optimum dalam tempoh masa yang memuaskan.

ABSTRACT

Software testing is a vital part in software development lifecycle. Most of the time, system under test has more than one input and testing of every combinations of inputs is almost impossible as the time of execution of test case is outrageously long. Combinatorial testing is the way to encounter exhaustive testing through the testing of every input values and every combination between parameters. Combinatorial testing can be divided into three types which are uniform strength interaction, variable strength interaction and input-output based relation (IOR). IOR combinatorial testing only test for the important combinations that selected by tester. Most of the researches in combinatorial testing applied uniform and variable interaction strength but there are only few studies feature IOR. Thus, IOR combinatorial testing is selected to be studied in this research. To overcome the combinatorial optimization problem, Jaya algorithm is proposed to apply in this project since metaheuristic algorithm is fast in optimization and this strategy is named as CTJ. The result of applying Jaya algorithm in input-output based combinatorial testing is acceptable since it produces nearly optimum number of test cases in the satisfactory time range.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACO	Ant Colony Optimization
AETG	Automatic Efficient Test Generator
AGC	Controller for Automatic Generation Control
AI	Artificial Intelligence
CA	Covering Array
EA	Evolutionary Algorithm
FIFT	Failure-Triggering Fault Interaction
GA	Genetic Algorithm
GUI	Graphical User Interface
HM	Harmony Memory
HMCR	Harmony Memory Considering Rate
HMS	Harmony Memory Size
HS	Harmony Search
IOR	Input-Output Based Relation
MCA	Mixed Level Covering Array
PAR	Pitch Adjustment Rate
PHSS	Pairwise Harmony Search Testing Strategy
PID	Proportional-Integral-Derivative
PSO	Particle Swarm Optimization
PSTG	Particle Swarm Test Generator
PV-DSTAT-COM	Photovoltaic Fed Distributed Static Compensator
QAP	Quadratic Assignment Problem
SA	Simulated Annealing
SBPWM	Simple Boost Pulse Width Modulation
SI	Swarm Intelligence
SUT	System Under Test
TLBO	Teaching-Learning-Based Optimization

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Software testing is an inevitable process in software development lifecycle to find out the software bugs by validating and verifying the application whether it works as expected and meets the business and technical requirements. A recent report from Tricentis, a leading software testing company in Continuous Testing found that there are 606 recorded software failure that happened around the globe which affected over 3.7 billion people and 314 companies as well as $1.7 trillion in lost revenue and 268 years of downtime (Tricentis, 2018). Therefore, a more effective defect detection approach needed to be carried out to increase the coverage of testing.

Combinatorial testing is a black-box testing technique that generate test cases by combining the values of different test object input parameters using combinatorial optimization strategies (De Vries, Vohra, Economics, & Science, 2003). Taking the study from the failure of medical device application, the failure-triggering fault interaction (FTFI) is 68% for single parameter value, 97% of failures triggered by 2 combination values while the percentage of failures caused by 3 and 4 combination values are 99% and 100% respectively (Kuhn, Wallace, & Gallo, 2004). By using combinatorial testing, all input values of the test objects and interactions between each parameter are tested which result in higher detection of interaction failure compared to single parameter testing.

Combinatorial optimization is a process of searching the optimum number of test cases for combinatorial testing. There are many different optimization strategies that are used to generate the test cases for combinatorial testing such as Harmony Search (A. R. A. Alsewari & Zamli, 2012), Genetic Algorithm (Shiba, Tsuchiya, & Kikuno, 2004b),

Ant Colony Algorithm (Shiba et al., 2004b), Simplified Swarm Optimization (Ahmed, Sahib, & Potrus, 2014), Differential Evolution Algorithm (Liang, Guo, Huang, & Jiao, 2014) and so on. Jaya Algorithm is chosen to be applied in this study as this algorithm has been used in lots of optimization problems in other fields.

Combinatorial testing also known as interaction t-way testing where t represents the interaction strength. There are two types of t-way interaction which are uniform strength t-way interaction and variable strength t-way interaction. The interaction between all parameters are uniform in uniform strength t-way interaction while variable strength t-way interaction involves main uniform interaction and sub-uniform interaction. Both type of interactions will generate all possible interactions between each parameter. Often, some of the interactions generated maybe not even be used in the testing. This waste the precious time and effort of the tester to generate those useless interactions. Hence, input-output based relation (IOR) has been introduced in combinatorial optimization to improve the efficiency in finding optimum number of test case as well as given the flexibility in selecting the desired parameter and its interaction (A. R. A. Alsewari & Zamli, 2012).

1.2 PROBLEM STATEMENT

In most of the software application, often there exists one part of system input required to enter a combination of values or choices. The system under test (SUT) is then needed to test for every combination of input parameter to make sure the actual behaviour of the system is same as expected behaviour since the cost of fixing the defect found after software delivered is much higher. Testing of each combination of values is a time and effort wasting job and this leads to exhaustive testing. Exhaustive testing is an impractical software testing technique and usually impossible to achieve in the real testing environment due to budget available and time constraint to execute all combinations of inputs.

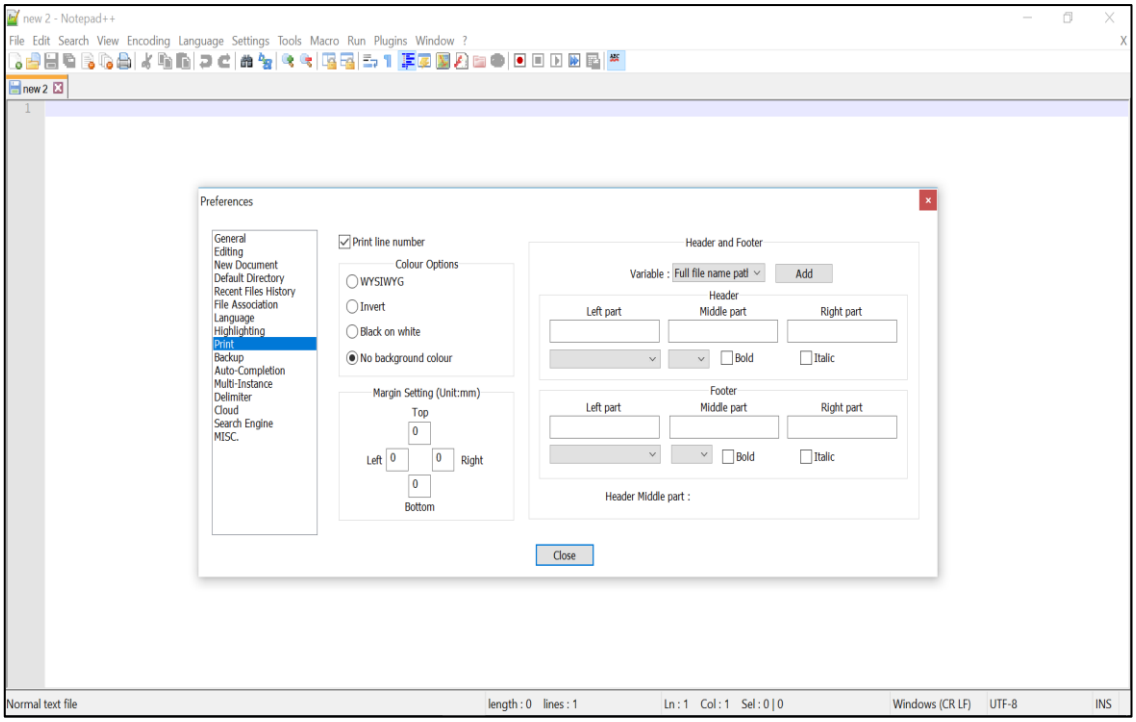

Figure 1.1 Print section of Preferences of Notepad ++

Taking the example from the renowned open source code editor, Notepad ++, the “Print” section in the Preferences as shown in Figure 1 is chosen to show the total number of test cases needed to carry out the testing process via exhaustive testing. There are 22 input parameters required to enter in the “Print” section and all possible input values are shown in Table 1.

REFERENCES

Abdul Rahman, A. A. (2012). A harmony search based pairwise sampling strategy for combinatorial testing. *International Journal of the Physical Sciences, 7*(7), 1062 - 1072. doi:10.5897/ijps11.1633

Abramson, D., & Abela, J. (1991). A parallel genetic algorithm for solving the school timetabling problem.

Ahmed, B. S., Sahib, M. A., & Potrus, M. Y. (2014). Generating combinatorial test cases using Simplified Swarm Optimization (SSO) algorithm for automated GUI functional testing. *Engineering Science and Technology, an International Journal, 17*(4), 218-226. doi:https://doi.org/10.1016/j.jestch.2014.06.001

Ahmed, B. S., & Zamli, K. Z. (2011). A variable strength interaction test suites generation strategy using Particle Swarm Optimization. *Journal of Systems and Software, 84*(12), 2171-2185. doi:https://doi.org/10.1016/j.jss.2011.06.004

Ahmed, B. S., Zamli, K. Z., & Lim, C. P. (2012). Constructing a T-Way Interaction Test Suite Using the Particle Swarm Optimization Approach. *International Journal of Innovative Computing, Information and Control, 8*(1), 431-452.

Alsewari, A. A., Tairan, N. M., & Zamli, K. Z. (2015). Survey on Input Output Relation based Combination Test Data Generation Strategies. *ARPN Journal of Engineering and Applied Sciences, 10*(18), 8427-8430.

Alsewari, A. A., & Zamli, K. Z. (2011). *Interaction Test Data Generation Using Harmony Search Algorithm.* Paper presented at the Proceeding of IEEE Symposium on Industrial Electronics & Applications, Langkawi, Malaysia.

Alsewari, A. R. A., & Zamli, K. Z. (2012). Design and implementation of a harmony-search-based variable-strength t-way testing strategy with constraints support. *Information and Software Technology, 54*(6), 553-568.

Ammar, M., Bouaziz, S., Alimi, A. M., & Abraham, A. (2013, 12-14 Aug. 2013). *Hybrid harmony search algorithm for global optimization.* Paper presented at the 2013 World Congress on Nature and Biologically Inspired Computing.

Arshem, J. (2009). TVG. Retrieved from http://sourceforge.net/projects/tvg

Blum, C. (2005). Ant colony optimization: Introduction and recent trends. *Physics of Life Reviews, 2*(4), 353-373. doi:https://doi.org/10.1016/j.plrev.2005.10.001

Blum, C., & Li, X. (2008). Swarm intelligence in optimization. In *Swarm intelligence* (pp. 43-85): Springer.

Bryce, R., & Colbourn, C. (2007). *One-Test-at-a-Time Heuristic Search for Interaction Test Suites.* Paper presented at the Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, London, England.

Bryce, R. C., & Colbourn, C. J. (2009). A Density-Based Greedy Algorithm for Higher Strength Covering Arrays. *Software Testing, Verification & Reliability, 19*(1), 37-53. doi:http://dx.doi.org/10.1002/stvr.v19:1

Chen, X., Gu, Q., Li, A., & Chen, D. (2009, 1-3 Dec. 2009). *Variable Strength Interaction Testing with an Ant Colony System Approach.* Paper presented at the 2009 16th Asia-Pacific Software Engineering Conference.

Chen, X., Gu, Q., Qi, J., & Chen, D. (2010). *Applying particle swarm optimization to pairwise testing.* Paper presented at the Computer Software and Applications Conference (COMPSAC), 2010 IEEE 34th Annual.

Cohen, D. M., Dalal, S. R., Parelius, J., Patton, G. C., & Bellcore, N. J. (1996). The Combinatorial Design Approach to Automatic Test Generation. *IEEE software, 13*(5), 83-88.

Cohen, M. B. (2004). *Designing Test Suites for Software Interaction Testing.* (Doctor of Philosophy PhD Thesis), University of Auckland, New Zealand.
Cohen, M. B., Colbourn, C. J., & Ling, A. C. H. (2003, 17-20 Nov. 2003). *Augmenting simulated annealing to build interaction test suites.* Paper presented at the 14th International Symposium on Software Reliability Engineering, 2003. ISSRE 2003.
Cohen, M. B., Gibbons, P. B., Mugridge, W. B., & Colbourn, C. J. (2003). *Constructing Test Suites for Interaction Testing.* Paper presented at the Proceedings of the 25th International Conference on Software Engineering, Portland, Oregon USA.
Colbourn, C. J., Cohen, M. B., & Turban, R. (2004). *A deterministic density algorithm for pairwise interaction coverage.* Paper presented at the IASTED Conf. on Software Engineering.
Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to algorithms third edition. *MIT Press. ISBN 0-262-03384-4. Section, 23*, 631-638.
De Vries, S., Vohra, R., Economics, N. U. C. f. M. S. i., & Science, M. (2003). Combinatorial auctions: A survey. *INFORMS Journal on Computing, 15*(3), 284-309.
Eberhart, R., & Kennedy, J. (1995). *A new optimizer using particle swarm theory.* Paper presented at the Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on.
Elbeltagi, E., Hegazy, T., & Grierson, D. (2005). Comparison among five evolutionary-based optimization algorithms. *Advanced Engineering Informatics, 19*(1), 43-53. doi:https://doi.org/10.1016/j.aei.2005.01.004
Geem, Z. W., & Kim, J. H. (2001). A New Heuristic Optimization Algorithm: Harmony Search. *Simulation, 76*(2), 60-68.
Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A New Heuristic Optimization Algorithm: Harmony Search. *Simulation, 76*(2), 60-68. doi:10.1177/003754970107600201
Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*: MIT press.
Ingber, L. (1993). Simulated annealing: Practice versus theory. *Mathematical and Computer Modelling, 18*(11), 29-57. doi:https://doi.org/10.1016/0895-7177(93)90204-C
Jenkins, B. (2003, February 5 2005). Jenny. Retrieved from http://burtleburtle.net/bob/math/jenny.html
Kennedy, J., & Eberhart, R. (1995). *Particle Swarm Optimization.* Paper presented at the Proceedings of IEEE International Conference Neural Networks.
Khazali, A. H., & Kalantar, M. (2011). Optimal reactive power dispatch based on harmony search algorithm. *International Journal of Electrical Power & Energy Systems, 33*(3), 684-692. doi:https://doi.org/10.1016/j.ijepes.2010.11.018
Kuhn, D. R., Wallace, D. R., & Gallo, A. M. (2004). Software fault interactions and implications for software testing. *IEEE Transactions on Software Engineering, 30*(6), 418-421. doi:10.1109/TSE.2004.24
Lam, S. S. B., Raju, M. L. H. P., M, U. K., Ch, S., & Srivastav, P. R. (2012). Automated Generation of Independent Paths and Test Suite Optimization Using Artificial Bee Colony. *Procedia Engineering, 30*, 191-200. doi:https://doi.org/10.1016/j.proeng.2012.01.851
Lei, Y., Kacker, R., Kuhn, D. R., Okun, V., & Lawrence, J. (2007). *IPOG: A General Strategy for T-Way Software Testing.* Paper presented at the Proceedings of the

14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, Tucson, AZ U.S.A.
Liang, X., Guo, S., Huang, M., & Jiao, X. (2014). Combinatorial Test Case Suite Generation Based on Differential Evolution Algorithm. *JSW, 9*(6), 1479-1484.
Mao, C., Yu, X., Chen, J., & Chen, J. (2012, 27-29 Aug. 2012). *Generating Test Data for Structural Testing Based on Ant Colony Optimization.* Paper presented at the 2012 12th International Conference on Quality Software.
McCaffrey, J. D. (2009, 20-24 July 2009). *Generation of Pairwise Test Sets Using a Genetic Algorithm.* Paper presented at the Proceedings of the 2009 33rd Annual IEEE International Computer Software and Applications Conference.
McCall, J. (2005). Genetic algorithms for modelling and optimisation. *Journal of Computational and Applied Mathematics, 184*(1), 205-222. doi:https://doi.org/10.1016/j.cam.2004.07.034
Mishra, S., & Ray, P. K. (2016). Power quality improvement using photovoltaic fed DSTATCOM based on JAYA optimization. *IEEE Transactions on Sustainable Energy, 7*(4), 1672-1680.
Ong, H. Y., & Zamli, K. Z. (2011). Development of Interaction Test Suite Generation Strategy with Input-Output Mapping Supports. *Scientific Research and Essays, 6*(16), 3418-3430. doi:Available online at http://www.academicjournals.org/SRE
Othman, R. R., & Zamli, K. Z. (2011). ITTDG: Integrated T-way Test Data Generation Strategy for Interaction Testing. *Scientific Research and Essays, 6*(17), 3638-3648. doi:Available online at http://www.academicjournals.org/SRE
Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization. *Swarm intelligence, 1*(1), 33-57.
Ramli, N., Othman, R. R., & Ali, M. S. A. R. (2016, 11-12 Aug. 2016). *Optimizing combinatorial input-output based relations testing using Ant Colony algorithm.* Paper presented at the 2016 3rd International Conference on Electronic Design (ICED).
Rao, R. (2016). Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *International Journal of Industrial Engineering Computations, 7*(1), 19-34.
Rao, R., More, K., Taler, J., & Ocłoń, P. (2016). Dimensional optimization of a micro-channel heat sink using Jaya algorithm. *Applied Thermal Engineering, 103*, 572-582.
Rao, R. V., Savsani, V. J., & Vakharia, D. (2011). Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-Aided Design, 43*(3), 303-315.
Schroeder, P. J. (2001). *Black-Box Test Reduction Using Input-Output Analysis.* (Ph.D. Ph.D.), Illinois Institute of Technology.,
Schroeder, P. J., Faherty, P., & Korel, B. (2002, 2002). *Generating expected results for automated black-box testing.* Paper presented at the Proceedings 17th IEEE International Conference on Automated Software Engineering.
Schroeder, P. J., & Korel, B. (2000). Black-box test reduction using input-output analysis. *SIGSOFT Softw. Eng. Notes, 25*(5), 173-177. doi:http://doi.acm.org/10.1145/347636.349042
Selvi, V., & Umarani, D. R. (2010). Comparative analysis of ant colony and particle swarm optimization techniques. *International Journal of Computer Applications (0975–8887), 5*(4).

Shiba, T., Tsuchiya, T., & Kikuno, T. (2004a, 28-30 Sept. 2004). *Using Artificial Life Techniques to Generate Test Cases for Combinatorial Testing.* Paper presented at the Proceedings of the 28th Annual International Computer Software and Applications Conference.

Shiba, T., Tsuchiya, T., & Kikuno, T. (2004b, 28-30 Sept. 2004). *Using artificial life techniques to generate test cases for combinatorial testing.* Paper presented at the Proceedings of the 28th Annual International Computer Software and Applications Conference, 2004. COMPSAC 2004.

Shin, K.-S., & Lee, Y.-J. (2002). A genetic algorithm application in bankruptcy prediction modeling. *Expert Systems with Applications, 23*(3), 321-328. doi:https://doi.org/10.1016/S0957-4174(02)00051-9

Singh, S. P., Prakash, T., Singh, V., & Babu, M. G. (2017). Analytic hierarchy process based automatic generation control of multi-area interconnected power system using Jaya algorithm. *Engineering Applications of Artificial Intelligence, 60*, 35-44.

Srivastava, P. R., & Kim, T.-h. (2009). Application of genetic algorithm in software testing. *International Journal of software Engineering and its Applications, 3*(4), 87-96.

Stardom, J. (2001). *Metaheuristics and the search for covering and packing arrays*: Simon Fraser University.

Tricentis. (2018). *Software Fail Watch: 5th Edition*. Retrieved from https://www.tricentis.com/wp-content/uploads/2018/02/20180207_Software-Fails-Watch.pdf

Vesterstrom, J., & Thomsen, R. (2004). *A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems.* Paper presented at the IEEE Congress on Evolutionary Computation.

Wang, Z., Xu, B., & Nie, C. (2008, 12-13 Aug. 2008). *Greedy Heuristic Algorithms to Generate Variable Strength Combinatorial Test Suite.* Paper presented at the 2008 The Eighth International Conference on Quality Software.

Wang, Z. Y., Xu, B. W., & Nie, C. H. (2008). *Greedy Heuristic Algorithms to Generate Variable Strength Combinatorial Test Suite.* Paper presented at the Proceedings of the 8th International Conference on Quality Software.

Warid, W., Hizam, H., Mariun, N., & Abdul-Wahab, N. I. (2016). Optimal power flow using the Jaya algorithm. *Energies, 9*(9), 678.

Wu, H., Nie, C., Kuo, F. C., Leung, H., & Colbourn, C. J. (2015). A Discrete Particle Swarm Optimization for Covering Array Generation. *IEEE Transactions on Evolutionary Computation, 19*(4), 575-591. doi:10.1109/TEVC.2014.2362532

Xambre, A. R., & Vilarinho, P. M. (2003). A simulated annealing approach for manufacturing cell formation with multiple identical machines. *European journal of operational research, 151*(2), 434-446.

Xiang, L. Y., Alsewari, A. A., & Zamli, K. Z. (2015). Pairwise test suite generator tool based on harmony search algorithm (HS-PTSGT). *International Journal on Artificial Intelligence, 2*.

Yu-Wen, T., & Aldiwan, W. S. (2000). *Automating Test Case Generation for the New Generation Mission Software System.* Paper presented at the Proceedings of the IEEE Aerospace Conference, Big Sky, MT, USA.

Ziyuan, W., Changhai, N., & Baowen, X. (2007). *Generating combinatorial test suite for interaction relationship.* Paper presented at the Proceeding of the 4th

international workshop on Software quality assurance: in conjunction with the 6th ESEC/FSE joint meeting, Dubrovnik, Croatia.